

新版软件步步高

基

础

本

Visual J++ 6.0

基础教程

捷新工作室 编著



- 最新 Java 集成开发工具
- 强大的 Microsoft Visual Studio 6.0 开发环境的重要成员
- 大量的设计向导可加快开发速度
- 与 Visual C++ 相似的界面风格
- 配有大量程序实例，易于掌握
- 内容基础，由浅入深

国防工业出版社

· 新版软件步步高(基础本)

Visual J++ 6.0 基础教程

捷新工作室 编著

国防工业出版社

100-17A
25

图书在版编目(CIP)数据

Visual J++ 6.0 基础教程/捷新工作室编著. - 北京:
国防工业出版社, 1999.4
(新版软件步步高·基础本)
ISBN 7-118-02061-3

I.V... II.捷... III.JAVA 语言 IV.TP312

中国版本图书馆 CIP 数据核字(1999)第 00574 号

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京怀柔新华印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 25½ 584 千字

1999 年 4 月第 1 版 1999 年 4 月北京第 1 次印刷

印数:1—4000 册 定价:34.00 元

(本书如有印装错误,我社负责调换)

总 序

在信息时代,知识成为推动社会生产力发展的一个最重要的因素,知识经济的轮廓在一些发达国家已经出现。以知识为基础的知识经济,其重要特点是信息产业的迅猛发展和产业的信息化,其内核是创新。我国是发展中国家,应该更加重视运用世界最新技术成果,有重点有选择地引进先进技术,增强自主创新能力,逐步实现技术发展的跨越。

在充分估量未来科学技术,特别是以计算机技术为先导的高技术发展对综合国力、社会经济结构和人民生活的巨大影响的基础上,为实现科教兴国战略多干实事,国防工业出版社组织了由数十位在计算机应用开发第一线工作的年富力强的博士、硕士组成的捷新工作室,编写出版《新版软件步步高(基础本)》和《新版软件步步高(提高本)》丛书。前者重在普及,后者追求提高,两者结合起来,力图满足多层次读者的需要。

《新版软件步步高(基础本)》的出版目的是普及新版软件的基本知识、基本操作技术,主要面向初学者,包括初次涉足该领域的机关、公司、企事业单位技术人员、大专院校师生及各类培训班学员,也可作为广大再就业职工理想的培训和学习教材。另外,对新技术感兴趣的读者也可将本丛书作为了解世界最新技术的窗口。

《新版软件步步高(提高本)》的出版目的是为有一定基础的读者找到提高专业技术水平和最新软件开发、操作技术的最佳途径,主要面向中高级读者,包括对该软件有一定基础知识,又希望提高自己专业技术水平的机关、公司、企事业单位技术人员、大专院校师生及各类高级培训班学员。

丛书的共同特点是突出一个“新”字,强调一个“精”字,力争一个“快”字。“新”是指软件的版本新;“精”是指精选的国内外流行最广、叫得最响的新版软件;“快”是指在保证质量的前提下,实现周期短,面市快。

丛书的内容覆盖最新高级语言开发环境(Visual J++ 6.0, Visual Basic 6.0, Visual Fox-Pro 6.0, Delphi 4.0),微机操作系统(中文 Windows 98, 中文 Windows NT 5.0),图表图像处理软件(Photoshop 5.0, 中文 CorelDRAW 8.0),Internet 浏览器(Internet Explorer 5.0),Web 页面设计环境(FrontPage 98),Internet 开发环境(Visual InterDev 6.0),大型数据库客户端开发工具(PowerBuilder 6.0)等。它们都是 1998 年以来推出的最新版软件。同时,我们还将把握新技术的脉搏,适时充实新的内容。

我们相信,丛书的出版必将为广大读者开辟跟踪、掌握、运用、进而创造新技术的最佳途径。

由于时间仓促,书中疏漏之处,敬请广大读者指正。

前 言

Java 是一种与平台无关的编程语言,可以开发 Internet(互联网)上散布的应用程序。它是由 SUN 公司的 JavaSoft 分部开发的一种划时代的新型编程语言。它是一种网络语言,适合编写像 Internet 这样的广域网上运行的各种应用程序,使原先静止不动的网页在 Java 中因拥有了动画、行为而变得灵活、安全,且充满生气。

Java 的用途并不仅此。Java 的独特功能使得向数百万的网上用户发送面向对象的应用程序成为可能。随着计算机产业的网络时代的到来,Java 已经宣告了基于网络计算的新纪元的到来。与其它任何一种语言开发工具不同,Java 是唯一适合在 Internet 上分布执行的语言。人们可以从 WWW(万维网)上的服务器动态下载 Java 程序,并且既可以在支持 Java 的浏览器上执行,也可以单独执行。

Java 作为面向 Internet 的编程语言,一推出就倍受青睐。但是,作为一种编程语言,需要有功能强大的集成开发环境,才能快速、方便地开发出应用程序。Microsoft Visual J++ 6.0 正是这样的集成开发环境之一,它使用与 Visual C++ 6.0 完全相同的环境 Microsoft Developer Studio 98,使得许多用户学习起来比较方便。

本书是国防工业出版社推出的《新版软件步步高(基础版)》丛书之一,本书使用了较多的篇幅介绍 Java 语言的基础知识,初学 Java 语言的读者应该耐心阅读这些知识,打下良好的基础。有兴趣的读者还可以进一步参考本书的姊妹篇《新版软件步步高(提高版)》丛书中的《Visual J++ 6.0 高级教程》。

内 容 简 介

本书全面地介绍了 Java 语言编程技术和微软最新的 Java 编程工具 Visual J++ 6.0。本书通过详细的图解和大量的实例介绍了怎样使用 Visual J++ 编写、调试、运行 Java 程序。本书内容包括 Java 技术综述、Java 语言语法、Java 语言的面向对象机制、包、接口和异常、输入/输出机制、HTML 页面、HTML 文字设计、表单、表和帧的设计、HTML 图像设计、HTML 增强标记功能、VBScript 语言基础、VBScript 变量和数组、VBScript 函数和过程、VBScript 程序流程控制、VBScript 客户端程序实例、Visual J++ 6.0 集成开发环境、文本域及按钮控件、文本区域和面板控件、Java 的布局设计、Java 的图形操作、Java 的用户界面设计、多线程技术；在附录中介绍了 Java 小程序编程原理、VBScript 语言参考和预定义颜色参考。

本书适用于大中专院校师生、公司职员及广大计算机爱好者，也可作为培训班教材。

目 录

第一章 Java 技术综述	1	2.3.1 声明一个变量	21
1.1 Java 语言概述	1	2.3.2 变量的动态初始化	22
1.1.1 Java 语言的特性	1	2.3.3 变量的作用域和生命期	22
1.1.2 Java 与 C++ 的区别	2	2.3.4 类型自动转换和强制转换	25
1.1.3 Java 语言的开发环境	3	2.3.5 表达式的自动类型提升	26
1.1.4 Java 的应用程序和小程序	4	2.4 数组类型	28
1.1.5 Java 程序的开发和发行机制	6	2.4.1 一维数组	28
1.1.6 Java 虚拟机概述	7	2.4.2 多维数组	30
1.1.7 Java 小程序与 HTML	7	2.4.3 其它数组声明句法	32
1.2 Java 技术的机理	8	2.4.4 指针:一个过时的概念	32
1.2.1 Java 机理	8	2.5 运算符	33
1.2.2 Java 实时系统	9	2.5.1 算术运算符	33
1.2.3 网络浏览器	10	2.5.2 位运算符	35
1.3 Java 安全性原理	10	2.5.3 关系运算符	35
1.3.1 安全级别	11	2.5.4 “?”运算符	36
1.3.2 Java 小程序的信任机制	12	2.5.5 运算符优先级	37
1.3.3 Authenticode	12	2.6 控制语句	37
第二章 Java 语言语法	14	2.6.1 选择语句	37
2.1 Java 语言的数据类型	14	2.6.2 迭代语句	44
2.1.1 强类型的语言	14	2.6.3 跳转语句	50
2.1.2 基本数据类型	14	第三章 Java 语言的面向对象机制 ..	54
2.1.3 整数类型	15	3.1 OOP 原理	54
2.1.4 浮点数类型	16	3.1.1 两种编程范式	54
2.1.5 字符类型	17	3.1.2 抽象	54
2.1.6 布尔型	18	3.1.3 三个基本原则	55
2.1.7 网址资源	19	3.2 Java 中的类	55
2.2 常量类型	20	3.2.1 类的一般定义形式	56
2.2.1 整数常量	20	3.2.2 一个简单的类实例	56
2.2.2 浮点常量	20	3.2.3 方法	59
2.2.3 布尔常量	20	3.2.4 构造函数	62
2.2.4 字符常量	20	3.2.5 this	63
2.2.5 字符串常量	21	3.2.6 重载	63
2.3 变量类型	21	3.2.7 把对象作为参数	67
		3.2.8 返回对象	68

3.2.9 递归	69	5.2.1 一般文件输入/输出	109
3.2.10 访问控制	70	5.2.2 FileIODemo	110
3.2.11 static	71	5.2.3 缓冲输入/输出	112
3.2.12 final	73	5.2.4 DataIODemo	114
3.2.13 数组的进一步讨论	73	5.3 文件与目录	116
3.3 Java 的继承机制	74	5.3.1 File	116
3.3.1 继承的概念	74	5.3.2 FileDemo	118
3.3.2 访问控制	76	5.4 网址	120
3.3.3 一个简单的继承实例	77	5.4.1 概述	120
3.3.4 super	79	5.4.2 InetAddress	121
3.3.5 方法重载	80	5.4.3 InetAddressDemo	122
3.3.6 动态方法派遣	83	5.5 URL	123
3.3.7 abstract	85	5.5.1 概述	123
3.3.8 有关 final 的进一步讨论	87	5.5.2 格式	123
第四章 Java 包、接口和异常	89	5.5.3 URL	124
4.1 包	89	5.5.4 URLEdemo	124
4.1.1 定义包	89	5.5.5 URLConnection	125
4.1.2 访问保护	90	5.5.6 UCDemo	125
4.1.3 import	90	5.6 TCP/IP Socket	126
4.2 接口	91	5.6.1 概述	126
4.2.1 定义一个接口	91	5.6.2 Socket	127
4.2.2 实现接口	92	5.6.3 SocketDemo	128
4.2.3 接口中的变量	93	5.6.4 ServerSocket	129
4.2.4 接口继承	95	5.7 UDP 和数据包	129
4.3 异常问题	96	5.7.1 概述	129
4.3.1 异常处理	96	5.7.2 DatagramPacket	130
4.3.2 异常类型	97	5.7.3 DatagramDemo	130
4.3.3 未捕获的异常	97	第六章 HTML 页面	133
4.3.4 使用 try 和 catch	98	6.1 HTML 简介	133
4.3.5 显示异常的描述	99	6.1.1 什么是 HTML	133
4.3.6 多个 catch 语句	99	6.1.2 用什么来编写 HTML 代码?	133
4.3.7 throw	100	6.1.3 HTML 编辑器	133
4.3.8 throws	101	6.1.4 一些基本概念	134
4.3.9 finally	102	6.2 HTML 页面布局	135
第五章 Java 输入/输出机制	105	6.2.1 HTML 的基本结构	135
5.1 控制台输入/输出	105	6.2.2 定制页面的属性	137
5.1.1 System 类	105	6.2.3 <BODY> 标记中的属性	139
5.1.2 ConsoleDemo	106	6.2.4 创建超链接	140
5.1.3 实现 main	107	6.2.5 开一个新的(浏览器)窗口	141
5.2 文件输入/输出	109	6.2.6 水平线	141
		6.2.7 有关颜色的设定	142

6.3 HTML 文本布局	144	10.3 HTML 的 < OBJECT > 标记 和 < PARAM > 标记	183
6.3.1 段落和换行	144	10.3.1 < OBJECT > 标记的一般 格式	184
6.3.2 创建列表	144	10.3.2 < OBJECT > 标记中的属性	184
6.3.3 预定的文本格式	148	10.3.3 < PARPM > 标记及其属性	185
6.3.4 其它与文本有关的标记	149	10.4 使用 < OBJECT > 标记 和 < PARAM > 标记	186
第七章 HTML 文字设计	153	10.5 HTML 多媒体页面设计 标记	188
7.1 HTML 字体设计	153	10.5.1 EMBED 标记	188
7.1.1 控制字体的大小	153	10.5.2 背景音乐	188
7.1.2 特殊效果的字体	154	10.5.3 插入视频剪辑	188
7.2 HTML 移动文字(MARQUEE) 设计	158	第十一章 VBScript 语言基础	190
7.2.1 什么是 MARQUEE	158	11.1 回顾 HTML 的 < SCRIPT > 标记	190
7.2.2 HTML 中的 < MARQUEE > 标记 及其属性	158	11.1.1 使用 < SCRIPT > 标记	190
第八章 表单、表和帧的设计	162	11.1.2 < SCRIPT > 标记的位置	192
8.1 HTML 表单(Form)设计	162	11.2 访问 VBScript 数据类型	192
8.1.1 创建和配置表单	162	11.3 判断变体中的数据类型	194
8.1.2 CGI 的基本认识	163	11.3.1 使用 VarType 函数	194
8.1.3 创建表单的 HTML 标记	163	11.3.2 使用 TypeName 函数	195
8.2 HTML 表(Table)设计	168	11.4 数据类型转换	195
8.2.1 表的基本概念	168	第十二章 VBScript 变量和数组	198
8.2.2 表的 HTML 标记简介	168	12.1 声明变量	198
8.2.3 表的 HTML 标记示例	170	12.2 变量的作用范围	198
8.3 HTML 帧(Frame)的设计	172	12.3 禁止使用未声明的变量	200
8.3.1 关于帧和帧集(Frame Set)	172	12.4 数组操作	200
8.3.2 创建帧的 HTML 标记	173	12.4.1 使用静态数组	200
第九章 HTML 图像设计	176	12.4.2 使用动态数组	202
9.1 在页面中插入图像	176	12.4.3 多维数组简介	203
9.2 在页面中使用图像的 HTML 标记简介	177	第十三章 VBScript 函数和过程	204
9.3 客户端图像映射图(Client Side Image Map)	178	13.1 VBScript 内建函数简介	205
第十章 HTML 增强标记功能	180	13.2 VBScript 日期和时间函数	205
10.1 HTML 的 < SCRIPT > 标记	180	13.3 通过消息框和输入框与 用户交互	206
10.1.1 < SCRIPT > 标记的属性	180	13.4 声明自定义函数	208
10.1.2 一小段 JavaScript 程序	180	13.5 声明定制过程	210
10.2 HTML 的 < APPLET > 标记	182		

第十四章 VBScript 程序流程控制 .. 212	第十九章 Java 的布局设计 296
14.1 用 If 和 Select 进行条件 判断 212	19.1 BorderLayout 布局 296
14.2 用 Do...Loop 在程序中 进行循环 213	19.2 CardLayout 布局 301
14.3 使用 For...Next 循环 215	19.3 GridLayout 布局 307
14.4 使用 While...Wend 循环 216	19.4 GridBagLayout 布局 313
第十五章 VBScript 客户端程序	第二十章 Java 的图形操作 321
实例 217	20.1 Java 的图形坐标系统 321
15.1 响应客户端事件 217	20.2 基本绘图方法 321
15.1.1 在 HTML 标记中定义事件 句柄 217	20.2.1 画线 321
15.1.2 在 VBScript 中定义事件处理 子例程 218	20.2.2 绘制矩形 322
15.2 对用户进入站点和离开站点 作响应 219	20.2.3 绘制椭圆 325
15.3 改变窗口中的 HTML 文档 .. 220	20.2.4 绘制圆弧 326
15.4 利用客户端脚本对 Form 元素进行验证 222	20.2.5 绘制多边形 327
第十六章 Visual J++ 6.0	20.2.6 复制图形 328
集成开发环境 225	20.3 颜色 329
16.1 Visual J++ 6.0 窗口界面 225	20.3.1 构造方法 329
16.2 创建一个 Java 小程序 226	20.3.2 常用方法 329
16.3 分析小程序 229	20.3.3 颜色常量 329
16.4 用模板创建 Java 小程序 231	20.3.4 设置颜色 330
16.5 Applet1.java 源程序 233	20.4 字体 331
16.6 HTML 简介 240	20.4.1 构造方法 332
16.7 解决方案和项目 243	20.4.2 常用方法 332
第十七章 文本域及按钮控件 247	20.4.3 类 Font 中定义的常量 332
17.1 文本域控件 247	20.4.4 设置字体 332
17.2 按钮控件 255	20.4.5 类 FontMetrics 333
17.3 Java 事件 260	第二十一章 Java 的用户界面
第十八章 文本区域和面板控件 273	设计 335
18.1 文本区域控件 273	21.1 AWT 包概述 335
18.2 面板控件 280	21.2 常用组件类编程 336
18.3 创建面板类 286	21.2.1 类 Label 336
	21.2.2 类 Button 338
	21.2.3 类 List 340
	21.2.4 类 TextArea 和类 TextField 343
	21.2.5 类 Checkbox 和类 CheckboxGroup 346
	21.2.6 类 Choice 349
	21.3 容器类 351
	21.3.1 类 Container 351
	21.3.2 类 Frame 351

第二十二章 多线程技术	354	A.2 小程序安全性模型	362
22.1 线程概述	354	A.3 小程序的生存周期	363
22.1.1 使用多线程的好处	354	A.4 小程序中的重要方法	363
22.1.2 使用多线程的时机	354	附录 B VBScript 语言参考	364
22.1.3 多线程的缺点	355	B.1 VBScript 数据子类型	364
22.2 一个多线程示例	355	B.2 VBScript 运算符	365
22.3 多线程的创建	356	B.3 VBScript 函数	368
22.3.1 继承类 Thread	356	B.4 VBScript 语句	378
22.3.2 实现接口 Runnable	360	B.5 VBScript 常数	383
附录 A Java 小程序编程概述	362	B.6 VBScript 编码规则	387
A.1 小程序的代码特点	362	附录 C 预定义颜色	390

第一章 Java 技术综述

虽然 Java 语言出现的时间很晚,但是它已经是 Internet 上最流行的编程语言了。Java 语言是一种完全的面向对象编程语言,它的语法类似于 C 和 C++ 语言,但是取消了 C 和 C++ 的一些复杂和容易混淆的语义特性。

1.1 Java 语言概述

Java 语言有许多特性使它非常适合于 Internet 编程,本节将介绍 Java 语言的特性,通过对这些特性的介绍,读者就会对 Java 语言有一个初步的了解。

1.1.1 Java 语言的特性

对于 C 和 C++ 程序员来说,学习 Java 语言是一件比较轻松的事,因为它的语法与 C 和 C++ 很类似,而且它比 C++ 要更简单一些。对于没有学过 C 和 C++ 编程的人来说,Java 也并不难掌握,因为它没有指针、运算符重载、多重继承等复杂的概念(这些概念在 C++ 中存在)。

Java 语言有很多特性,下面我们就介绍它最主要的特性:平台独立性、安全性、面向对象的特性。

1. 平台独立性

我们都知道,运行在 UNIX 平台上的程序是无法直接运行在 Windows 环境下的,这是因为它们所基于的硬件和操作系统不同。所以,要想使一个在 UNIX 平台上非常优秀的软件能够运行在 Windows 平台上,就需要开发人员对这种软件进行移植。而移植的代价通常相当大,不仅会耗费大量的人力物力,而且很可能使这个在原来运行平台上非常不错的软件变得平庸。

如果软件能够独立于平台,也就是说可以不经修改地在各种平台上(例如 UNIX、Windows NT 等)运行,那么将是一件多么值得高兴的事情,因为我们的可用软件会成倍地增加。用 Java 语言编写的软件就具有这种平台独立性。

Java 语言是一种解释执行的语言,但是它与传统的解释执行语言(例如 Basic 语言)有所不同。在用 Java 语言编写完程序之后,首先把 Java 源代码编译成一种特殊的独立于平台的机器码,也就是 Java 字节码。然后由 Java 虚拟机(Virtual Machine)负责解释执行字节码。我们只需要针对各种平台编写各种虚拟机,就可以使字节码在各种平台上由各自的虚拟机解释执行,从而实现了平台独立性。

此外,为保证平台独立性,Java 语言还采用了其它一些措施。例如,使用统一的数据类型标准,即在各种平台上数据类型是一致的。Java 语言的整数类型 int 表示 32 位整数,

无论在微机还是工作站上,也不管操作系统是 Windows 3.1 还是 Windows NT,它都是表示 32 位整数,不会随着平台不同而改变整数的表示位数。

2. 安全性

网络安全性一直是非常令人头痛的问题,尤其是在 Internet 上,如果用来编程的语言没有良好的安全性,那么就会造成很大的损失。例如通过网络购物时,用户输入了自己的信用卡号,如果没有良好的网络安全性,那么信用卡号就有可能被窃取。

Java 语言在设计时就考虑了程序的安全性问题,主要有以下几个方面的措施:

首先 Java 语言取消了 C/C++ 中的极其重要的指针类型。毫无疑问,指针类型是一种非常有用的数据类型,但是由于 Java 语言把自己定位于一种网络编程语言,因此安全性才是一个最值得关注的问题。取消了指针可以防止用户非法地访问程序之外的内存,避免了恶意用户对计算机的侵害。

在执行 Java 字节码时,Java 语言的虚拟机还要对字节码进行安全检查。因为字节码在编译生成后到执行前有可能被非法改变,所以在执行之前还要使用字节码校验器对字节码进行有效性验证。只有通过验证的 Java 字节码才能运行。

在运行时,Java 字节码被严格限制在特定的内存区域内,任何对这个内存区域外的访问都是被禁止的,甚至对于一些基本资源(例如本地的文件)的访问都被禁止了(是不是太严格了?)。

3. 面向对象特性

Java 语言是一种 100% 的面向对象的语言。为什么是 100% 呢? 因为 Java 语言中所有的数据和方法(在 Java 中,我们把函数叫做方法)都是定义在类中的,也就是说,Java 语言的数据和方法不能脱离于对象而存在。

面向对象语言有一些共同的特性,例如继承性、多态性等等,这些特性 Java 语言当然也都具备,有关这方面的内容我们将在介绍 Java 类时详细的讲述。

1.1.2 Java 与 C++ 的区别

Java 语言在设计初期曾经考虑把 C++ 语言作为蓝本来进行设计,后来虽然由于种种原因而放弃了这种想法,但是在 Java 语言中还是处处有 C++ 的影子。尤其是一些基本的语法规则,例如每条语句以分号(;)结束,使用花括号({})作为语句段的起始和结束标志等等都与 C++ 相同。正是由于这些非常相似的特性,使得 C++ 程序员可以比较容易地掌握 Java 语言。

不过在看到这些相似的特点时,不能忽略两者之间的差别。对于 C/C++ 程序员,掌握两者之间的差别比了解它们的共性更重要。

下面我们就来了解一些两者之间的区别:

1. 适用范围的差别

任何一种语言都有自己的定位。Java 语言给自己的定位就是网络编程,具体地说就是 Internet/Intranet 编程;而 C++ 则是一种通用性很强的语言,在几乎各种应用中都可以使用 C++ 语言进行编程。

2. 可移植性的差别

Java 语言具有平台独立性,即实现了软件的跨平台可执行。而 C++ 虽然理论上具有

较好的可移植性,但是由于存在多种不同的开发环境(Borland 的 C++ Builder、Microsoft 的 Visual C++),所以给移植带来了困难。

3. 安全性的差别

Java 语言在设计时就考虑了安全性问题,通过取消指针等不安全的因素,使得用 Java 语言编写的程序本身就具有一定的安全性。与之相比,C++ 在安全性方面没有过多的考虑。

1.1.3 Java 语言的开发环境

任何一种语言都需要有一个开发环境,Java 也不例外。最早推出 Java 语言开发环境的也就是 Java 语言的发明者——SUN 公司。这套基于命令行方式的开发环境被称为 Java Developers Kit(JDK)。

虽然后来又有许多公司推出了各种开发工具(例如 Microsoft 的 Visual J++, Symantec 的 Cafe),但是都离不开最基本的 JDK。所以下面我们就来介绍一下 Java JDK。

1. JDK 的组成

SUN 公司为多种平台提供了 JDK,在这里我们介绍 Windows 95 平台上的 JDK。

JDK 主要是由编译器、解释器和浏览器组成的,当用户在本地计算机上安装完 JDK 后,在 JDK 目录下就会出现一个名为 bin 的目录,在这个目录中就包含编译器、解释器和小程序浏览器。

假设 Java JDK 安装在了 C:\java 目录中,那么在目录 C:\java\bin 中就会包含 javac.exe、java.exe 和 appletviewer.exe。

2. JDK 的编译器

编译器 javac.exe 是用 Java 语言编写的。它的功能就是把 Java 源程序(后缀名为 .java)编译成 Java 字节码文件(后缀名为 .class)。编译器 javac.exe 可以同时编译多个源程序,并输出 .class 文件;源程序中的每个 Java 类,编译器都会对应生成一个同名的 .class 文件。

编译器的命令行格式如下所示:

javac [编译选项] 文件 1,文件 2...

Java 的编译器有几个编译选项,通过它们可以指定编译 Java 源程序的方式。表 1.1 列出了这些编译选项。

表 1.1 Java 编译器的编译选项

编译选项	简要说明
- CLASSPATH < path >	引用类的路径表
- d < directory >	编译后 .class 文件存放的目录
- g	编译时生成调试信息表
- ng	编译时不生成调试信息
- nowarn	禁止编译器的警告功能
- o	在编译时使用优化功能,使生成的 .class 文件具有更高的执行速度
- verbose	显示编译过程的详细信息

3. JDK 的解释器

JDK 的解释器 `java.exe` 的功能是解释执行 Java 字节码(即 `.class` 文件),该解释器的命令行格式如下所示:

```
java [解释选项] 字节码文件 [参数表]
```

在这里指定的字节码文件必须带有一个方法 `main()`,这个方法就是程序的入口点。同样,该方法可以带有参数表。

4. 小程序浏览器

在 JDK 中包含一个小程序浏览器 `appletviewer`,这个简单的浏览器是为了快速检验小程序而提供的。因为 Java 小程序只能运行在浏览器中(例如, Netscape 的 Navigator、Microsoft 的 Internet Explorer),所以如果每次都运行它们来观看小程序的执行结果,那么就不太方便。而小程序浏览器由于功能简单,启动迅速,用它来观看小程序的运行结果将非常方便。

有关 Java 小程序的概念我们将在 1.1.4 节介绍。

1.1.4 Java 的应用程序和小程序

在使用 Java 语言编程时,可以生成两种类型的程序:一种是 Java 应用程序(Java application),另一种是 Java 小程序(Java applet),这两种程序从结构上、执行的方式上来说都有比较大的区别。

Java 应用程序与一般的可独立执行的程序相同,用户可以直接执行它。而 Java 小程序虽然与 Java 应用程序有相似之处,但是 Java 小程序不能直接执行,必须通过支持 Java 的 WWW 浏览器(例如 Navigator 和 Internet Explorer)来执行。因此,Java 小程序实际上是浏览器中执行的程序。

下面我们就通过例子来说明它们之间的差别。

1. Java 应用程序

我们可以在任何一种文本编辑器中编写 Java 应用程序,例如在 Windows 中常用的记事本(NotePad)中。如下所示,程序清单 1.1 是一个经典的 Hello, world! Java 程序示例:

程序清单 1.1

```
//程序 HelloWorldApp.java
class HelloWorldApp{
    public static void main(String args[]){
        System.out.println("Hello, world!");
    }
}
```

这个程序可能是最简单的 Java 应用程序了,它只实现输出 Hello, world! 的功能。语句 `System.out.println` 是 Java 为程序开发人员提供了 Java API。

Java API,即 Java Application Programming Interface,是 Java 提供的各种程序包和类的集合。

编写完上面的程序后,就可以使用前面介绍的 Java 编译器编译这个程序,如下所示:

```
javac HelloWorldApp.java
```

如果编译器顺利完成了编译工作,那么就会在当前的目录下生成一个名为 HelloWorldApp.class 的文件。

现在就可以执行编译生成的 HelloWorldApp.class 了,如下所示我们使用 Java JDK 的解释器 java 来执行这个字节码文件:

```
java HelloWorldApp
程序将在屏幕上输出结果:
Hello, world!
```

2. Java 小程序

与编写 Java 应用程序的方法相同,用户也可以在任何一个文本编辑器中编写 Java 小程序,如下所示,程序清单 1.2 就是一个 Java 小程序的例子:

程序清单 1.2

```
//程序 HelloWorldApplet.java
import java.awt.Graphics;
import java.applet.Applet;

public class HelloWorldApplet extends Applet{
    public void init(){
        resize(100,100);
    }

    public void paint(Graphics g){
        g.drawString("Hello, world!",20,20);
    }
}
```

这个程序同样是输出 Hello,world!,但是与 Java 应用程序有明显的区别。

首先,程序使用了两个 import 语句引入了两个类。Java 中的 import 语句的功能与 C 语言中 include 语句很相似。第一个 import 语句引入了类 Graphics,其前缀 java.awt 指出这个类包含在程序包 awt 中;第二个 import 语句引入类 Applet。

接着,定义类 HelloWorldApplet,因为这是一个小程序,所以这个类必须是类 Applet 的派生类,其中关键字 extends 用来表示类 HelloWorldApplet 派生于类 Applet。

在类 HelloWorldApplet 中有两个方法 init 和 paint,其中方法 init 用来初始化小程序,方法 paint 用来输出 Hello,world!。

按照上面介绍的程序清单 1.2 编写完后,就可以编译这个小程序,编译的方法与编译 Java 应用程序相同:

```
javac HelloWorldApplet.java
```

这个小程序编译通过后,将会产生一个类文件 HelloWorldApplet.class。

前面我们介绍过,Java 小程序必须通过浏览器来执行。为此 JDK 专门提供了 appletviewer 来执行小程序。不过,appletviewer 并不能直接载入小程序,还要通过一个 HTML 文档来载入小程序,如下所示,就是一个 HTML 文档的示例:

程序清单 1.3

```

< HTML >
< HEAD >
< TITLE > Lello world < /TITLE >
< /HEAD >
< APPLET CODE = "HelloWorldApplet.class" WIDTH = 100 HEIGHT = 100 >
< /APPLET >
< /HTML >

```

在任何一种文本编辑器中输入这个 HTML 文档后,保存该文档,并给其命名为 Hello.html。

使用 appletviewer 执行该小程序:

```
appletviewer HelloWorldApplet.html
```

这时屏幕上就会显示一个窗口,窗口中显示程序输出的"Hello, world!"。

1.1.5 Java 程序的开发和发行机制

Java 开发平台及相关支持技术的更新速度是非常快的。商业 Java 程序在 Java 语言中的开发和部署很好的说明了 Java 的快速成熟。这主要是由以下的因素造成的:

(1) 降低了开发时间和花费。Java 语言一开始就是面向对象的,这样,开发者一旦掌握了它面向对象的特点,开发工作能够比常规的循序渐进方法进行得更有效。而且,Java 语言在快速生成高级和强壮的软件方面比 C++ 有更大的优越性。

(2) 使用商业类库。Java 语言有大量的内置类。这些类是开发者编写 Java 程序的基础。渐渐的,商业类库的出现使开发者的工作有一个很高的起点。例如,如果开发者能够使用提供用户界面功能(如各种窗口控制)的商业类库,他编写 Java 小程序的工作将简单得多。

(3) 使用基于组件的应用程序。很多软件商认为 Java 是新一代可繁殖程序的基础,这些可繁殖程序是基于对象或组件编程模型的。在这个模型中,应用程序是有一些可下载的组件组成的,而不是一个单一的闭门造车的产物。

在使用一个基于组件的应用程序时,用户并不需要将整个程序安装在他的本地系统上。相反,用户可以只从浏览器上下载他所需要的那部分功能。例如,在创建一个字处理程序时,用户从第一个小程序中得到主要的字处理功能,而图形功能可以在需要时才从网上的第二个小程序中下载下来。

这种程序开发的组件模型和传统的软件开发方式有很大的不同,现在还只处于初级阶段。如果程序开发员采用这种开发模型,他将面对很大的挑战。

Java 语言和 Java 实时系统允许开发者在万维网(WWW 网)上发行 Java 小程序。小程序一般都很小,在浏览器提出请求时可以被下载并执行。Java 小程序的加入给万维网增色不少。例如,Java 小程序可以提供高级控制并对用户产生的事件作出反应。Java 语言提供丰富的事件模型,它比 HTML 具有更强大的功能。

Java 还允许开发者编写可以脱离网页浏览器运行的有特色的 Java 独立程序。在这方面,Java 有可能代替 C 语言或其它的语言,这是因为 Java 具有以下特点:

- (1) 使用 Java 强大的网络功能可以改善许多本地基于网络的客户/服务器程序的性能。
- (2) Java 独立程序可以在一个组织的本地网上开发、安装和运行,而 Java 小程序没有